

## Лабораторная работа 2.2

### Элементы информационно-управляющих систем — работа с параллельным портом ПК.

**Цель работы:** Знакомство с алгоритмами работы простых информационно-управляющих систем, работа с параллельным портом в ОС Linux.

**Аппаратное и программное обеспечение:** PC, ОС Linux с установленным компилятором GCC, подключенный к параллельному порту сегментный индикатор.

#### Порядок выполнения работы:

1. Работа выполняется на локальной или удаленной машине с ОС Linux и установленным компилятором GCC. При выполнении работы на удаленной машине создайте рабочий каталог с именем выполняющих работу (команда **mkdir Ivanov\_Petrova**).
2. Необходимо создать файл lab22.c содержащий код программы управления внешним устройством через параллельный порт ПК (LPT). В качестве внешнего устройства используется сегментный индикатор, подключенный к LPT порту по схеме, показанной на рисунке 2.2.1.

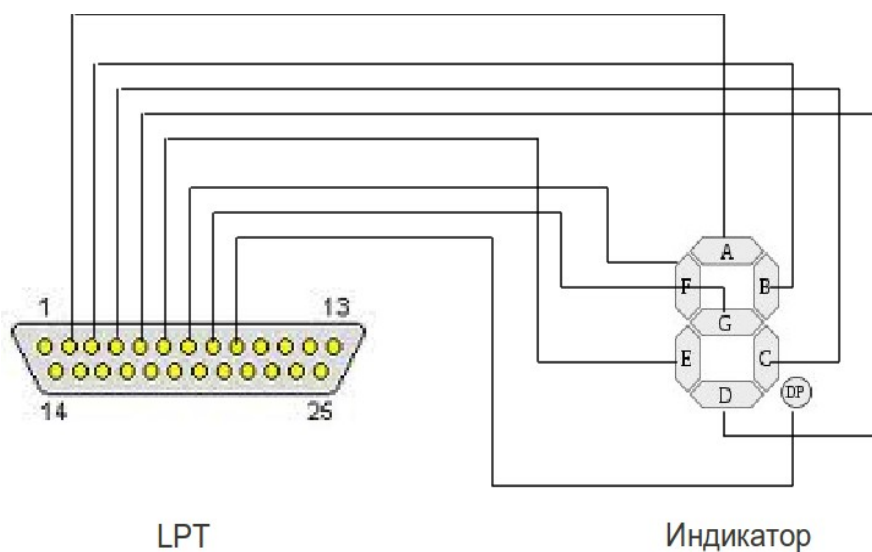


Рисунок 2.2.1 Схема подключения сегментного индикатора.

3. Выводы 2-9 параллельного порта являются выходами и используются для управления индикатором. Сегмент индикатора загорается при выставлении логической «1» на соответствующем выводе порта. Выставить логическую «1» или «0» можно, отправив на адрес порта LPT (например 0x378 для первого LPT порта) некоторое число. Например, если отправить в порт число 255 (11111111 в бинарном формате), то на всех выходах порта будут установлены «1». Пример кода:

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/io.h>
#include <sys/types.h>
#include <fcntl.h>
```

```

#define BASEPORT 0x378 /* lp1 */

int main() {
    char c;
    int n, tem;

    printf("Hit any key to stop\n");

    //set permissions to access port
    if (ioperm(BASEPORT, 3, 1)) {perror("ioperm"); exit(1);}

    tem = fcntl(0, F_GETFL, 0);
    fcntl (0, F_SETFL, (tem | O_NDELAY));

    //main loop where actual blinking is done
    while (1) {
        //if some key is pressed, break out from loop
        n = read(0, &c, 1);
        if (n > 0) break;

        //write 'on' bit on all data pins and wait 1/4 second
        outb(255, BASEPORT);
        usleep(250000);

        //write 'off' bit on all data pins and wait 1/4 second
        outb(0, BASEPORT);
        usleep(250000);
    }

    fcntl(0, F_SETFL, tem);
    outb(0, BASEPORT);

    //take away permissions to access port
    if (ioperm(BASEPORT, 3, 0)) {perror("ioperm"); exit(1);}

    exit(0);
}

```

4. На основе вышеприведенного кода составить программу, которая должна выводить на сегментный индикатор номера выполняющих работу (согласно списку группы). Номера выводятся следующим образом: через секунду после запуска программы выводится номер первого студента, выполняющего работу (сначала на индикаторе должен появиться разряд десятков, а через 0.25 секунды разряд единиц), далее через секунду аналогичным образом выводится номер второго выполняющего работу и так далее. Например, работу выполняют два студента с номерами в списке группы **03** и **14** соответственно, тогда на индикатор выводятся следующие цифры: (пауза 1 секунда) **0** (пауза 0.25 секунды) **3** (пауза 1 секунда) **1** (пауза 0.25 секунды) **4**, далее все повторяется с начала. Программа должна прерываться при нажатии любой клавиши на клавиатуре.
5. Для компилирования программы необходимо выполнить команду:

**gcc lab22.c -o lab22**

программа запускается в привилегированном режиме, команда запуска:

**sudo ./lab22**

**Источники информации:**

<http://www.google.ru>